ISO TC 184/SC4/WG12 N 527 Date: 00-05-23
Supersedes ISO TC 184/SC4/WG_12

ISO/ [ballot stage] [standard number] - [part number] - [ballot cycle]
Document title : JNC Proposal of STEP Assembly Model for Products
This document superposes ISO TC 184/WG12 N513

ABSTRACT:

Establishment of neutral assembly model within STEP is quite important and urgent issue from various application viewpoints, such as parametric assembly, assembly/disassembly process planning, kinematic analysis, and tolerance analysis. The objective of the assembly model presented in this document is to establish a neutral representation of assemblies of products, which are composed of sets of components.

KEYWORDS:
Component association, assembly feature, assembly feature association

COMMENTS TO READER:
This document is a quick version for making entities needed for describing assembly model consistent with the latest status of STEP 40 series parts.

OWNER/EDITOR: Nobuhiro SUGIMURA
Address: Osaka Prefecture University
1-1 Gakuencho, Sakai, Osaka 599-8531, Japan

TEL: +81-722-54-9207
FAX: +81-722-54-9904
E-mail: **sugimura@center.osakafu-u.ac.jp**

Alternate: Akihiko OHTAKA
Address: Nihon Unisys Co. Ltd.
1-1-1, Toyosu, kohtoh, Tokyo 135-8560, Japan

TEL: +81-3-5546-4784
FAX: +81-3-5546-7810
E-mail: akihiko.ohtaka@unisys.co.jp

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards in normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This International Standard is organized as a series of parts, each published separately. This part of ISO 10303 falls into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, application test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1. This part of ISO 10303 is a member of the integrated resource series.

This part of ISO 10303 is intended primarily to supplement the ISO 10303 40-seies parts to allow the representation of product models enhanced with assembly structure information.

The following items are required to represent an assembly composed of a set of components.

(1) Information about individual parts.
(2) Information about standard parts.
(3) Structure configuration of assembly
   a) Hierarchical associations (parent-child associations) among assemblies, subassemblies and piece parts.
   b) Positions and orientations of components in a higher level component.
   c) Tolerance of the positions and orientations.
(4) Component association
   a) Peer-to-peer associations among components.
   b) Relative positions and orientations of components against other components.
   c) Relative motions of components against other components.
   d) Tolerance of the relative motions, positions and orientations.
   e) Assembly features representing detailed information about the component association.

The information models dealing with the items (1), (2) and (3) have already been established in ISO 10303 and ISO 13584. Therefore, the assembly model deals with the following items of the assembled products.

Component association
a) Peer-to-peer associations among components.
b) Relative positions and orientations of components against other components.
c) Relative motions of components against other components.
d) Assembly features representing detailed information about the component association.

Industrial automation systems and integration – Product data representation and exchange – Part 1xx: Application resources: Assembly model

1. Scope

This part of ISO 10303 specifies the resources to describe the associations among the components of an assembled product.

The followings are within the scope of this part of ISO 10303.

- the connecting associations among the components constituting a assembled product;
- the associations among the components which are not physically connected;
- the relationships among the associations of the components;
- the description of the product composed of both the designed components and the standard components;
- the characteristic features of the associations among the components;
- the design, the analysis and the manufacturing preparation of the assembled products;

The following are outside of the scope of this part of ISO 10303

- the configuration management of the assemblies and the components;

2. Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 8824-1:1994, *Information technology – Open system interconnection – Abstract syntax notation one (ASN.1) – Part 1: Specification of basic notation*.

ISO 10303-1: 1994, *Industrial automation systems and integration – Product data representation and exchange – Part 1: Overview and fundamental principles*.

ISO 10303-11: 1994, *Industrial automation systems and integration – Product data representation and exchange – Part 11:Description methods: The EXPRESS language reference manual*.

ISO 10303-41: [1)], *Industrial automation systems and integration – Product data representation and exchange – Part 41: Integrated generic resource: Fundamentals of product description and support*.

ISO 10303-43: [1)], *Industrial automation systems and integration – Product data representation and exchange – Part 43: Integrated generic resource: Representation structure*.

ISO 10303-44: [1)], *Industrial automation systems and integration – Product data representation and exchange – Part 44: Integrated generic resource: Product structure configuration.*

ISO 10303-105: 1996, *Industrial automation systems and integration – Product data representation and exchange – Part 105: Integrated application resource: Kinematics.*

ISO/CD 10303-108: [1)], *Industrial automation systems and integration – Product data representation and exchange – Part 108: Integrated application resource: Parametrization and constraints for explicit geometric models.*

[1)] To be published

3. Definitions

3.1 Terms defined in ISO 10303-1

This part of ISO 10303 makes use of the following terms in ISO 10303-1.

- **Assembly**: a product that is decomposable into a set of components or other assemblies from the perspective of a specific application.
- **Component**: a product that is not subject to decomposition from the perspective of a specific application.

3.2 Terms defined in ISO 10303-44

This part of ISO 10303 makes use of the following terms in ISO 10303-44.

- **Sub-assembly**: a constituent that is an assembly.

3.3 Other definitions

For the purpose of this part of ISO 10303, the following definitions apply.

3.3.1 component association: an association between a pair of piece parts and/or subassemblies.

3.3.2 assembly feature: an element to specify the associations between a pair of piece parts and/or subassemblies.

4. Assembly_model_schema

The following *EXPRESS* declaration begins the assembly_model_shcema and identifies the necessary references.

EXPRESS specification:

```
*)
SCHEMA  assembly_model_schema;

REFERENCE FROM product_definition_schema
  ( product_definition,
    product_definition_relationship );
REFERENCE FROM product_property_definition_schema
  ( characterized_object,
    property_definition,
    product_definition_shape,
    shape_aspect,
    shape_aspect_relationship);
REFERENCE FROM product_property_representation_schema
  ( shape_definition_representation,
    shape_representation,
    represented_definition );
REFERENCE FROM support_resource_schema
  ( bag_to_set,
    identifier,
    label,
    text );
REFERENCE FROM geometry_schema
  ( geometric_representation_item );
REFERENCE FROM representation_schema
  ( using_representations,
    representation_item,
    representation,
    item_defined_transformation,
    representation_item_relationship,
    representation_relationship );
REFERENCE FROM product_structure_schema
  ( assembly_component_usage,
    next_assembly_usage_occurence );
REFERENCE FROM kinematic_structure_schema
  ( kinematic_pair );
REFERENCE FROM kinematic_motion_representation_schema
  ( kinematic_path );
REFERENCE FROM explicit_geometric_constraint_schema
  ( explicit_geometric_constraint );
REFERENCE FROM parametric_assembly_constraint_schema
  ( parametric_assembly_constraint );
(*
```

NOTE – The schemas referenced above can be found in the following parts of ISO 10303:

product_definition_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
support_resource_schema	ISO 10303-41
representation_schema	ISO 10303-43
product_structure_schema	ISO 10303-44
kinematic_structure_schema	ISO 10303-105
kinematic_motion_representation_schema	ISO 10303-105
explicit_geometric_constraint_schema	ISO 10303-108

parametric_assembly_constraints_schema ISO 10303-108

4.1 Introduction

The objective of the assembly model presented in this document is to establish a neutral representation of assemblies of products, which are composed of sets of components. The products to be described by the assembly model are summarized in the followings.

(1) products composed of sets of components.
The products considered here are the assembled products composed of sets of the components. The whole products are called "assemblies", and the components of the lowest levels in the assemblies are called "parts". The components of the intermediate levels are called "sub-assemblies", which are composed of one or more parts and/or sub-assemblies. An assembly consists of one or more sub-assemblies and parts.

(2) Product structure configuration of assembly
Product structure configuration is now dealt with in ISO 10303 - 44 to describe the parts lists and the BOM (bill-of-material). Various structure configurations are given to one assembly depending on various contexts. For instance, one configuration of an assembly is considered in the design phase, and the structure configuration may be changed in the assembly process planning phase. The objective of the assembly model is to establish a model describing both the product structure configuration and the connecting associations among the components needed in the various design, analysis and manufacturing process planning phases.

(3) Standard parts
The assemblies include many standard parts, such as fixing bolts, keys, and electric motors. The standard parts are basically divided into two; they are, the standard parts included in the parts catalogues discussed in ISO TC 184/SC4/WG2, and the standard parts defined by the users.

The interest application fields of the assembly model are as follows.
(1) Kinematic analysis of assemblies
Kinematic analysis and simulation are very important application fields of the assembly model. ISO 10303-105 Kinematics supports the kinematic analysis, however, Part 105 is not sufficient to integrate the 3D-CAD systems and the kinematic analysis systems.

(2) Animation of assemblies
The animation of assemblies is very important for future extension of the digital-mockup technologies.

(3) Assembly/disassembly process planning

The assembly model will support the integration of the product design and the manufacturing preparation. The assembly process planning and the disassembly process planning are important application fields of the assembly model from the viewpoint of the integration of CAD and CAM systems.

(4) Tolerance analysis and synthesis

The tolerance analysis and synthesis of the complicated assembly are very important application fields of the assembly model.

4.2 Fundamental concept and assumptions

Some examples related to the fundamental concept and assumptions mentioned here are described in the Annex E of this document.

4.2.1 Hierarchical structure of assembly

The target products to be represented by the assembly model consist of three types of definitions, which are the subtype of the **product_definition** defined in Part 41 in this standard.

(1) assembly definition

The **assembly_definition**s are the **product_definition**s that represent the whole target products.

(2) subassembly definition

The **subassembly_definition**s are the **product_definition**s that represent the intermediate components of the assemblies.

(3) piece part definition

The **piece_part_definition**s are the **product_definition**s that represent the piece parts of the assemblies. The piece parts mean the components which can not be subdivided into any components.

The hierarchical associations among the **assembly_definition**s, the **subassembly_defintion**s and the **piece_part_defition**s are represented by the **next_assembly_usage_occurrence** defined in Part 44 in this standard. Two subtypes shown in the followings are defined as the subtype of **the next_assembly_-occurrence**.

(a) main component usage

The **main_component_usage** is the **next_assembly_usage_occurence** that represents the association between a parent **product_definition** (an assembly or a subassembly) and a child **product_definition** (another subassembly or a piece part) that is a main and important component of the parent.

(b) auxiliary component usage

The **auxiliary_component_usage** is the **next_assembly_usage_occurence** that represents the association between a parent **product_definition** (an assembly or a subassembly) and a child **product_definition** (another subassembly or a piece part) that is an auxiliary component of the parent. For example, standard parts, such as bolts, nuts and rivets, are typical examples of the auxiliary components.

4.2.2 Associations among components

The components constituting the assemblies have some kinds of peer to peer associations. For example, two piece parts are physically connected with each other. The **component_association**, a subtype of the **product_definition_relationship** in Part 44 in this standard, is defined to represent the peer to peer associations between a pair of **product_definition**s.

The **component_association** has three subtypes based on the types of the associations between a pair of **product_definition**s.

(1) connection

The **connection** is an entity which represents the physical connections between a pair of components. This entity is applied to describe the physical connections between a pair of components, such as a joint between a pair of components.

(2) relative motion

The **relative_motion** is an entity which represents the relative motions between a pair of components which are not physically connected with each other. This entity is applied to describe the constraints on the relative motions between a pair of components. A relative motion of a robot hand against a base of the robot is a typical example of the relative motion. In this case, the robot hand is not connected directly with the base.

(3) relative position and orientation

The **relative_position_and_orientation** is an entity which represents the relative positions and orientations between a pair of components which are not physically connected with each other. This entity is applied to describe the constraints on the relative position and orientation of the component against another component.

An entity named **component_association_relationship** is defined to represent the relationship between a pair of **component_association**s, and is classified into following two subtypes.

(a) component association hierarchy

The **component_association_hierarchy** is an entity which represents the parent-child relationship (hierarchical relationship) between a pair of **component_association**s. For example, a **component_association** between a pair of subassemblies is a parent **componet_association** of another **component_association** between a pair of parts included in these subassemblies.

(b) component association alternative

The **component_association_alternative** is an entity which represents the interchangeable relationships between a pair of **component_association**s. For example, if two **component_-association**s defined for a same pair of components can be interchangeable, these two **component_association**s should be associated each other by this entity.

4.2.3 Assembly features and their association

The assembly features are the elemental entities for representing the **component_association**s between a pair of **product_definition**s. For example, a hole and a cylinder are typical assembly features which represent the physical connection between a bearing part and a shaft part. An entity named **assembly_feature**, a subtype of **shape_aspect** defined in Part 41 in this standard, is defined to present these assembly features.

These features are associated in the assembly model, and an entity named **assembly_feature_-association** is defined to represent the associations between a pair of **assembly_feature**s. For example, the **assembly_feature** of the hole and the one of the cylinder are connected by the **assembly_feature_-association**, when the bearing part and the shaft part are assembled.

4.2.4 Representation of assembly feature associations

The **assembly_feature_association** should be represented by some elements of the representation schema in Part 43 in this standard. An entity named **relationship_identified_representation_usage** is defined to connect the definition entities of **shape_aspect**, **shape_aspect_reltionship** and **property_definition** with the representation entities of **representation_item_relationship**, **representation_relationship** and **item_defined_transformation**.

An entity named **assembly_feature_association_representation** is defined as a subtype of **relationship_identified_representation_usage**, which specifies the constraints on the relationships between the **assembly_feature_association**s and their representations.

4.3 assembly_model_schema type definition: representing_relationhip

An **representing_relationship** is a selection of types of relationships between pair of **representation_item**s and **representation**s.

EXPRESS specification:

```
*)
TYPE representing_relationship = SELECT
  ( representation_item_relationship,
    representation_relationship,
    item_defined_transformation);
END_TYPE; --representing_relationship
(*
```

4.4 assembly_model_schema entity definitions

4.4.1 assembly_definition

A subtype entity of the **product_definition**, which describes the definition of the assemblies composed of sets of parts and subassemblies.

EXPRESS specification:

```
*)
ENTITY assembly_definition
  SUBTYPE OF ( product_definition );
WHERE
  WR1: SIZEOF (ancestor_product_definition([], [SELF])) = 0;
END_ENTITY; --assembly_definition
 (*
```

Formal propositions:
WR1: the **assembly_definition** is the highest level instance in the tree structure defined by the **assembly_component_usage.**

4.4.2 subassembly_definition

A subtype entity of the **product_definition**, which describes the definition of the subassemblies composed of sets of parts and other subassemblies.

EXPRESS specification:

```
*)
```

```
ENTITY subassembly_definition
  SUBTYPE OF ( product_definition );
WHERE
  WR1: (SIZEOF (ancestor_product_definition([], [SELF])) <> 0 ) AND
       (SIZEOF (descendant_product_definition([], [SELF])) <> 0 );
END_ENTITY; --subassembly_definition

(*
```

Formal propositions:
WR1: the **subassembly_definition** is not any terminal instances in the tree structure defined by the **assembly_component_usage.**

4.4.3 piece_part_definiton

A subtype entity of the **piece_part_definition**, which describes the definition of the individual parts.

EXPRESS specification:

```
*)
ENTITY piece_part_definition
  SUBTYPE OF ( product_definition );
WHERE
  WR1: SIZEOF (descendant_product_definition([], [SELF])) = 0;
END_ENTITY; --piece_part_definition
(*
```

Formal propositions:
WR1: the **piece_part_definition** is the lowest level instance in the tree structure defined by the **assembly_component_usage**.

4.4.4 main_components_usage

An entity which represents the parent-child associations between a pair of components. This entity is a subtype of **next_assembly_usage_occurrence** in Part 44. The attribute **relating_product_definition** of this entity represent a parent **product_definition**, which is an assembly. The attribute **related_product_definition** of this entity specifies the main components of the parent component.

EXPRESS specification:

```
*)
ENTITY main_components_usage
  SUBTYPE OF ( next_assembly_usage_occurence );
  auxiliary_components : SET [0:?] OF auxiliary_components_usage;
END_ENTITY; --main_components_usage
(*
```

Attribute definitions:
auxiliary_components: the set of **auxiliary_component_usage**, which specify the parent-child associations between the parent product definition and the auxiliary component definitions.

4.4.5 auxiliary_components_usage

An entity which represents the parent-child associations between a pair of components. The attribute **relating_product_definition** of this entity represent a parent **product_definition**, which is an

assembly. The attribute **related_product_definition** of this entity specifies the auxiliary components of the parent. The auxiliary components keep the association among the main components of the parent **product_definition**. When two body panels of an automotive are connected by a se of bolts, the bolts are the auxiliary component and the panels are the main components, respectively.

EXPRESS specification:

```
*)
ENTITY auxiliary_components_usage
  SUBTYPE OF ( next_assembly_usage_occurence );
END_ENTITY; --auxiliary_components_usage
(*
```

4.4.6 component_association

A supertype entity which represents the peer to peer associations between a pair of components. This entity is a subtype of the **product_definition_reltionship**. The associations considered here are one to one (binary) associations between a pair of components.

EXPRESS specification:

```
*)
ENTITY component_association
  SUBTYPE OF ( product_definition_relationship );
WHERE
  WR1: SELF\product_definition.relationship.relating_product_definition :<>:
       SELF\product_definition_relationship.related_product_definition;
  WR2: (('ASSEMBLY_MODEL_SCHEMA.ASSEMBLY_DEFINTION' IN TYPEOF(
        SELF\product_definition_relationship.relating_product_definition)) OR
        ('ASSEMBLY_MODEL_SCHEMA.SUBASSEMBLY_DEFINTION' IN TYPEOF(
        SELF\product_definition_relationship.relating_product_definition)) OR
        ('ASSEMBLY_MODEL_SCHEMA.PIECE_PART_DEFINTION' IN TYPEOF(
        SELF\product_definition_relationship.relating_product_definition))) AND
       (('ASSEMBLY_MODEL_SCHEMA.ASSEMBLY_DEFINTION' IN TYPEOF(
        SELF\product_definition_relationship.related_product_definition)) OR
        ('ASSEMBLY_MODEL_SCHEMA.SUBASSEMBLY_DEFINTION' IN TYPEOF(
        SELF\product_definition_relationship.related_product_definition)) OR
        ('ASSEMBLY_MODEL_SCHEMA.PIECE_PART_DEFINTION' IN TYPEOF(
        SELF\product_definition_relationship.related_product_definition)));
END_ENTITY; --component_association
(*
```

Formal propositions:
WR1: the **relating_product_definition** and the **related_product_definition** should be different instances.
WR2: both the **relating_product_definition** and the **related_product_definition** specify **assembly_definition**, **subassembly_definition**, or **piece_part_definition**.

4.4.7 component_association_property

An entity which represents the property about the **component_association**.

EXPRESS specification:

```
*)
ENTITY component_association_property
```

```
  SUBTYPE OF ( property_definition );
  SELF\property_definition.definition : component_association;
END_ENTITY; --component_association_property
(*
```

Attribute definitions:
definition: the **component_association** to which the property is given.

Note: The contents of the properties will be discussed and developed.

4.4.8 component_shape_association

This entity specify the association between the **component_association**s and the **shape_aspect_relationship**s. The **shape_aspect_relationship**s represent the detailed information about the **component_association**s. If two components are connected by two pairs of the holes and the pegs, for example, the **shape_aspect_relationship**s represent the individual hole-peg relationship, and the **component_association** specifies the association between the two components.

EXPRESS specification:

```
*)
ENTITY component_shape_association;
  id : identifier;
  name : label;
  description : text;
  definition_relationship : product_definition_relationship;
  aspect_relationship : shape_aspect_relationship;
UNIQUE
  UR1: id,
       definition_relationship,
       aspect_relationship;
WHERE
  WR1: (SELF.definition_relationship.relating_product_definition IN
         using_product_definition_of_shape_aspect
         (SELF.aspect_relationship.relating_shape_aspect)) AND
       (SELF.definition_relationship.related_product_definition IN
          using_product_definition_of_shape_aspect
          (SELF.aspect_relationship.related_shape_aspect));
END_ENTITY; --component_shape_association
(*
```

Attribute definitions:
id: the identification of the **component_shape_association**.
name: the word or group of words by which the **component_shape_association** is referred to.
description: text that relates the nature of the **component_shape_association**.
definition_relationship: the **product_definition_relationship** which is associated with the **shape_aspect_relationship**.
aspect_relationship: the **shape_aspect_relationship** which represent the detailed information about the **product_definition_ralationship**.

Formal propositions:
UR1: The **id**, **definition_relationship** and **aspect_relationship** uniquely identify an instance of **component_shape_association**.
WR1: the **relating_shape_aspect**s of the **shape_aspect_relationship** shall be a portion of the **ralating_product_definition** of the **product_definition_relationship**. The **related_shape_aspect**s of the **shape_aspect_relationship** shall be a portion of the **ralated_product_definition** of the

product_definition_relationship.

4.4.9 connection

A supertype entity which represents the connections between a pair of components which are physically connected with each other. This entity is applied to describe the physical connections between a pair of components.

EXPRESS specification:

```
*)
ENTITY connection
  SUBTYPE OF ( component_association );
END_ENTITY; --connection
(*
```

4.4.10 movable_connection

An entity which represents the associations between a pair of components which are physically connected and movable. This entity is applied to describe the possible relative motions between a pair of components and the properties of the joints, which constrain the components. Typical examples of the **movable_connection** are the shaft-bearing joints, slider-guide way joints, gear joints, and so on.

EXPRESS specification:

```
*)
ENTITY movable_connection
  SUBTYPE OF ( connection );
END_ENTITY; --movable_connection
(*
```

4.4.11 fixed_connection

An entity which represents the connections between a pair of components which are physically connected and fixed. This entity is applied to describe the properties of the joints, which fix the components with each other. Typical examples of the **fixed_connection** are the welded joints, the key fastenings, the screw fastenings, and so on.

EXPRESS specification:

```
*)
ENTITY fixed_connection
  SUBTYPE OF ( connection );
END_ENTITY; --fixed_connection
(*
```

4.4.12 intermittent_connection

An entity which represents the connections between a pair of components which are physically connected with each other intermittently. This entity is applied to describe the physical interfaces between the intermittently connected components, such as limit switches.

EXPRESS specification:

```
*)
ENTITY intermittent_connection
```

```
  SUBTYPE OF ( connection );
END_ENTITY; --intermittent_connection
(*
```

4.4.13 relative_motion

An entity which represents the relative motions between a pair of components which are not physically connected with each other. This entity is applied to describe the constraints on the relative motions between a pair of components. A relative motion of a robot hand against a base of the robot is a typical example of the relative motion. In this case, the robot hand is not connected directly with the base.

EXPRESS specification:

```
*)
ENTITY relative_motion
  SUBTYPE OF ( component_association );
END_ENTITY; --relative_motion
(*
```

4.4.14 relative_position_and_orientation

An entity which represents the relative positions and orientations between a pair of components which are not physically connected with each other. This entity is applied to describe the constraints on the relative position and orientation of the component against another component.

EXPRESS specification:

```
*)
ENTITY relative_position_and_orientation
  SUBTYPE OF ( component_association );
END_ENTITY; --relative_position_and_orientation
(*
```

4.4.15 component_association_relationship

A supertype entity which represents the relationship between a pair of **component_association**s.

EXPRESS specification:

```
*)
ENTITY component_association_relationship;
  id : identifier;
  name : label;
  description : text;
  relating_component_association : component_association;
  related_component_association : component_association;
UNIQUE
  UR1: id,
       relating_component_association,
       related_component_association;
WHERE
  WR1: relating_component_association :<>: related_component_association;
END_ENTITY; --component_association_relationship
(*
```

Attribute definitions:
id: the identification of the **component_association_relationship**.
name: the word or group of words by which the **component_association_relationship** is referred to.

description: text that relates the nature of the **component_association_relationship**.
relating_component_association: one of the **component_association** which is a part of relationship.
relatied_component_association: the other **component_association** which is a part of relationship.

Formal propositions:
UR1: The **id**, **relating_component_association** and **related_component_association** uniquely identify an instance of **component_association_relationship**.
WR1: the **relating_component_association** and the **relatied_component_association** should be different instances.

4.4.16 component_association_hierarchy

An entity which represents the parent-child relationship (hierarchical relationship) between a pair of **component_association**s. For example, the **component_association** between a pair of subassemblies is a parent **componet_association** of the **component_association** between a pair of parts included in these subassemblies. The **relating_component_association** is the parent association, and the **related_component_association** is the child association.

EXPRESS specification:

```
*)
ENTITY component_association_hierarchy
  SUBTYPE OF ( component_association_relationship );
WHERE
  WR1: SELF\component_association_relationship.relating_component_association\
       product_definition_relationship.relating_product_definition IN
       ancestor_product_definition
       ([], SELF\component_association_relationship.related_component_association\
        product_definition_relationship.relating_product_definition);
  WR2: SELF\component_association_relationship.relating_component_association\
       product_definition_relationship.related_product_definition IN
       ancestor_product_definition
       ([], SELF\component_association_relationship.related_component_association\
        product_definition_relationship.related_product_definition);
END_ENTITY; --component_association_hierarchy
(*
```

Formal propositions:
WR1: the **relating_component_association.relating_component_definition** shall be a higher level **product_definition** (sub-assembly or assembly) of the **relatied_component_association.relating_-product_definition** in the tree structures which are specified by the **assembly_component_usage**.
WR2: the **relating_component_association.related_component_definition** shall be a higher level **product_definition** (sub-assembly or assembly) of the **relatied_component_association.related_-product_definition** in the tree structures which are specified by the **assembly_component_usage**.

4.4.17 component_associatoion_alternative

An entity which represents the interchangeable relationships between a pair of **component_-association**s. For example, if two **component_association**s defined for a same pair of components can be interchangeable, these two **component_association**s should be associated each other.

EXPRESS specification:

```
*)
ENTITY component_association_alternative
  SUBTYPE OF ( component_association_relationship );
```

```
WHERE
WR1:(SELF\component_association_relationship.relating_component_association\
          product_definition_relationship.relating_product_definition :=:
     SELF\component_association_relationship.related_component_association\
          product_definition_relationship.relating_product_definition) AND
    (SELF\component_association_relationship.relating_component_association\
          product_definition_relationship.related_product_definition :=:
     SELF\component_association_relationship.related_component_association\
          product_definition_relationship.related_product_definition);
END_ENTITY; --component_association_alternative
(*
```

Formal propositions:
WR1: Two **component_association**s, which are related by a **component_association_alternative**, should describe the **component_associations** between a same pair of the **product_definition**s.

4.4.18 feature_definition

This entity represents the definitions of the feature definitions which are the portions of the product definitions.

EXPRESS specification:

```
*)
ENTITY feature_definition
  SUBTYPE OF ( shape_aspect );
END_ENTITY; --feature_definition
(*
```

4.4.19 assembly_feature

This entity represents the assembly features from the viewpoint of the application fields of the assembly model. The assembly features are the key elements for describing the **shape_aspect**s or the **geometric_represetaion_item**s on which the components are associated with each other. The assembly feature may be the partial shape elements of the components, on which two components are associated with each other.

EXPRESS specification:

```
*)
ENTITY assembly_feature
  SUBTYPE OF ( characterized_object, feature_definition );
END_ENTITY; --assembly_feature
(*
```

4.4.20 assembly_feature_property

An entity that represents the property of the **assembly_feature**.

EXPRESS specification:

```
*)
ENTITY assembly_feature_property
  SUBTYPE OF ( property_definition );
  SELF\property_definition.definition : assembly_feature;
END_ENTITY; --assembly_feature_property
(*
```

Attribute definitions:
definition : the **assembly_feature** to which the property is given.

Note: The contents of the properties will be discussed and developed.

4.4.21 assembly_feature_association

This entity represents the associations between pairs of assembly features from the viewpoint of the application fields of the assembly model. The **assembly_feature_association**s are the key elements for describing the associations between a pair of the components. The assembly feature association may specify a pair of assembly features, on which two components are associated with each other.

EXPRESS specification:

```
*)
ENTITY assembly_feature_association
  SUBTYPE OF ( shape_aspect_relationship );
  SELF\shape_aspect_relationship.relating_shape_aspect: assembly_feature;
  SELF\shape_aspect_relationship.related_shape_aspect: assembly_feature;
UNIQUE
  UR1: id,
       SELF\shape_aspect_relationship.relating_shape_aspect,
       SELF\shape_aspect_relationship.related_shape_aspect;
WHERE
  WR1: relating_assembly_feature :<>: related_assembly_feature;
END_ENTITY; --assembly_feature_association
(*
```

Attribute definitions:
relating_shape_aspect: one of the **assembly_feture** which is a part of relationship.
related_shape_aspect: the other **assembly_feture** which is a part of relationship. If one element of the relationship depends upon the other, this attribute shall be the dependent one.

Formal propositions:
UR1: The **id**, **relating_shape_aspect** and **related_shape_aspect** uniquely identify an instance of **assembly_feature_association**.
WR1: two of the **assembly_feature**s assiciated by this entity shall be different instances.

4.4.22 assembly_feature_association_property

An entity that represents the property of the **assembly_feature_association**.

EXPRESS specification:

```
*)
ENTITY assembly_feature_association_property
  SUBTYPE OF ( property_definition );
  SELF\property_definition.definition : assembly_feature_association;
END_ENTITY; --assembly_feature_association_property
 (*
```

Attribute definitions:
definition : the **assembly_feature_association** to which the property is given.

Note: The contents of the properties will be discussed and developed.

4.4.23 relationship_identified_representation_usage

This entity identifies a relationship between a pair of r**epresentation**s or between a pair of **representation_item**s as being the element that describes a particular component or part of the property.

EXPRESS specification:

```
*)
ENTITY relationship_identified_representation_usage;
  name : label;
  description : text;
  definition : represented_definition;
  representation_relation : representing_relationship;
END_ENTITY; --relationship_identified_representation_usage
(*
```

Attribute definitions:
name: the word or group of words by which the **relationship_identified_representation_usage** is referred to.
description: text that relates the nature of the **relationship_identified_representation_usage**.
definition: the identification of the **general_property property_definition**, **property_definition_-reltionship**, **shape_aspect**, or **shape_aspect_relationship** that is represented.
representation_relation: the identification of the **representation_relationship**, .**representation_item_-relationship**, or **item_defined_transformation** that represent the property.

Note: The contents of the properties will be discussed and developed.

4.4.24 assembly_feature_association_representation

An entity that represents the property of the **assembly_feature_association**.

EXPRESS specification:

```
*)
ENTITY assembly_feature_association_representation
  SUBTYPE OF ( relationship_identified_representation_usage );
UNIQUE
  UR1: SELF\relationship_identified_representation_usage.definition,
       SELF\relationship_identified_representation_usage.representation_relation;
WHERE
  WR1: 'ASSWMBLY_MODEL_SCHEMA.ASSEMBLY_FEATURE_ASSOCIANTION' IN TYPEOF
       (SELF\relationship_identified_representation_usage.definition);
  WR2: ('PARAMETRIC_ASSEMBLY_CONSTRAINTS_SCHEMA.PARAMETRIC_ASSEMBLY_CONSTRAINT'
        IN TYPEOF (SELF\relationship_identified_representation_usage.
                   representation_relation)) OR
       ('KINEMATIC_STRUCTURE_SCHEMA.KINEMATIC_PAIR' IN TYPEOF
        (SELF\relationship_identified_representation_usage.
                   representation_relation)) OR
       ('REPRESENTATION_SCHEMA.REPRESENTATION_RELATIONSHIP_WITH_MOTION' IN TYPEOF
        (SELF\relationship_identified_representation_usage.
                   representation_relation));
  WR3: consistent_of_af_association_and_af_association_rep(SELF);
END_ENTITY; --assembly_feature_association_representation
(*
```

Attribute definitions:

SELF\elationship_identified_representation_usage.definition : the **assembly_feature_association** to which the property is given.

Formal propositions:
UR1: The inherited attributes, the **definition** and the **representation relation,** uniquely identify an instance of **assembly_feature_association_representation**.
WR1: the inherited attribute **definition** identifies the **assembly_feature_association**.
WR2: the inherited attribute representation_relation identifies the **parmetric_assembly_constraint**, **kinameic_pair**, or **representation_relationship_with_motion**.
WR3: the **relating_shape_aspect** of the **assembly_feature_association** is represented by the **relating_representation** or **the related_representation_item** of the relating element of the **representing_relationship**, and also the **related_shape_aspect** of the **assembly_feature_association** is represented by the **related_representation** or **the related_representation_item** of the relating element of the **representing_relationship**.

4.4.25 representation_relationship_with_motion

This entity represents the relative motion of one **representation** against the other **representation**.

EXPRESS specification:

```
*)
ENTITY representation_relationship_with_motion
  SUBTYPE OF ( representation_relationship );
  motion : kinematic_path;
END_ENTITY; --representation_relationship_with_motion
(*
```

Attribute definitions:
motion : the **kinematic_path** which describe the relative motion of the **related_representation** against the **relating_representation**.

4.5 Assembly_model_schema function definitions

This subclause contains the EXPRESS function definitions in the **assembly_model_schama**.

4.5.1 ancestor_product_definition

The function **ancestor_product_definition** determines all the **product_definition** that are the ancestor of the specified **product_definitions** in the tree structure defined by the **product_structure_schema.-assembly_component_usage**.

EXPRESS specification:

```
*)
-- This function return all the product_definition that are ancestor of the
-- specified product_definition
FUNCTION ancestor_product_definition
  (ancestor: SET OF product_definition;
   child: SET OF product_definition) : SET OF product_definition;
LOCAL
  local_parent: SET OF product_definition := [];
  local_relation: SET OF assembly_component_usage := [];
  i : INTEGER := 0;
```

```
  j : INTEGER := 0;
END_LOCAL;

-- ERROR child is vacant --
IF (SIZEOF (child) = 0 ) THEN RETURN ([]);

-- extraction of related assembly_component_relationships –-
  ELSE
    REPEAT j:= 1 TO HIINDEX(child);
      local_relation := BAG_TO_SET (USEDIN (child[j],
                     'PRODUCTURE_STRUCTURE_SCHEMA.ASSEMBLY_COMPONENT_USAGE.'+
                     'RELATED_PRODUCT_DEFINITION'));
    END_REPEAT;
  IF (SIZEOF (local_relation) = 0) THEN RETURN (ancestor);
     ELSE

-- extraction of additional ancestor --
     REPEAT i :=1 TO HIINDEX(local_relation);
       REPEAT j := 1 TO HIINDEX(child);
         IF (local_relation[i].relating_product_definition <> child [j]) THEN
          local_parent := local_parent +
local_relation[i].relating_product_definition;
        END_IF;
       END_REPEAT;
     END_REPEAT;
     IF (SIZEOF (local_parent) = 0 ) THEN RETURN (ancestor);
       ELSE
         ancestor := ancestor + local_parent;

-- ERROR ancestor includes all product_definitions --
         IF (SIZEOF (ancestor) = HIINDEX (product_definition)) THEN RETURN
(ancestor);

-- extraction of higher level ancestors --
           ELSE ancestor := ancestor_product_definition (ancestor, local_parent);
         END_IF;
      END_IF;
   END_IF;
END_IF;
END_FUNCTION;
(*
```

Attribute definitions:

ancestor: the candidate set of **product_definition**s which shall be the ancestors of the child **product_definition**s in the tree structure defined by the **product_structure_schema.assembly_-component_usage**.

child: the input set of **product_definition**s. The ancestors of these **product_definition**s are extracted by this function.

4.5.2 descendant_product_definition

The function **descendent_product_definition** determines all the **product_definition** that are the descendents of the specified **product_definitions** in the tree structure defined by the **product_structure_schema.assembly_component_usage**.

EXPRESS specification:

```
*)
-- This function return all the product_definition that are descendant of the
specified
```

```
-- product_definition --
FUNCTION descendant_product_definition
  (descendant: SET OF product_definition;
   parent: SET OF product_definition) : SET OF product_definition;
LOCAL
  local_child: SET OF product_definition := [];
  local_relation: SET OF assembly_component_usage := [];
  i : INTEGER := 0;
  j : INTEGER := 0;
END_LOCAL;

-- ERROR parent is vacant --
IF (SIZEOF (parent) = 0 ) THEN RETURN ([]);

-- extraction of related assembly_component_relationships –-
  ELSE
    REPEAT j:= 1 TO HIINDEX(parent);
       local_relation := BAG_TO_SET (USEDIN (parent[j],
                     'PRODUCTURE_STRUCTURE_SCHEMA.ASSEMBLY_COMPONENT_USAGE.'+
                     'RELATING_PRODUCT_DEFINITION'));
    END_REPEAT;
  IF (SIZEOF (local_relation) = 0) THEN RETURN (descendant);
     ELSE

-- extraction of additional descendant --
     REPEAT i :=1 TO HIINDEX(local_relation);
       REPEAT j := 1 TO HIINDEX(parent);
         IF (local_relation[i].related_product_definition <> parent [j]) THEN
          local_child := local_child + local_relation[i].related_product_definition;
        END_IF;
       END_REPEAT;
     END_REPEAT;
     IF (SIZEOF (local_child) = 0 ) THEN RETURN (descendant);
       ELSE
         descendant := descendant + local_child;

-- descendant includes all product_definitions --
         IF (SIZEOF(descendant) = HIINDEX(product_definition)) THEN RETURN
(descendant);

-- extraction of higher level descendants --
           ELSE descendant := descendant_product_definition (descendant,
local_child);
         END_IF;
      END_IF;
   END_IF;
END_IF;
END_FUNCTION;
(*
```

Attribute definitions:
descendant: the candidate set of **product_definitions** which shall be the descendant of the parent **product_definition**s in the tree structure defined by the **product_structure_schema.assembly_-component_usage**.
parent: the input set of **product_definition**s. The descendant of these **product_definition**s are extracted by this function.

4.5.3 using_product_definition_of_shape_aspect

The function **using_product_definition_of_shape_aspect** determines all the **product_definition**s that

use the specified shape_aspect to define the shape of the **product_definition**s.

EXPRESS specification:

```
*)
-- This function extract product_definition using a shape_aspect
FUNCTION using_product_definition_of_shape_aspect
    (item:shape_aspect) : product_definition;
IF ('PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF
    (item.of_shape\property_definition.definition)) THEN
    RETURN (item.of_shape\property_definition.definition);
    ELSE RETURN ([]);
END_IF;
END_FUNCTION;
(*
```

Attribute definitions:
item: the input **shape_aspect** used to define the shape of the **product_definition**s, which shall be extracted.

4.5.4 using_product_definition_of_representation

The function **using_product_definition_of_representation** determines all the **product_-definition**s that use the specified **representation** to represent the shape of the **product_definition**s.

EXPRESS specification:

```
*)
-- This function extract a product_definitions using a geometric_representation
FUNCTION using_product_definition_of_representation
    (item: representation) : SET OF product_definition;
LOCAL
  local_s_d_r: SET OF shape_definition_representation := [];
  local_p_d_s: SET OF product_definition_shape := [];
  local_s_a: SET OF shape_aspect := [];
  local_p_d: SET OF product_definition := [];
  i : INTEGER;
END_LOCAL;

-- find shape_definition_representations
  local_s_d_r  := BAG_TO_SET (USEDIN (item,
                'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.SHAPE_DEFINITION_' +
                '_REPRESENTATION.REPRESENTATION_MODEL'));

-- find product_definition_shape & product_definitions
REPEAT i := 1 TO HIINDEX (local_s_d_r);
   IF (('PRODUCT_PROPERTY_DEFINITION_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF
       (local_s_d_r[i].representation_of)) AND
       ('PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION' IN
       (local_s_d_r[i].representation_of\property_definition.definition))) THEN
        local_p_d := local_p_d +
                    local_s_d_r[i].representation_of\property_definition.definition;
   ELSE IF (('PRODUCT_PROPERTY_DEFINITION_SCHEMA.SHAPE_ASPECT' IN TYPEOF
           (local_s_d_r[i].representation_of)) AND
           ('PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION' IN
        (local_s_d_r[i].representation_of.of_shape\property_definition.definition)))
   THEN local_p_d := local_p_d +
                    local_s_d_r[i].representation_of.of_shepe\
                    property_definition.definition;
   END_IF;
```

```
   END_IF;
END_REPEAT;
RETURN (local_p_d);
END_FUNCTION;
(*
```

Attribute definitions:

item: the input **representation** used to represent the shape of the **product_definition**s, which shall be extracted.

4.5.5 using_product_definition_of_item

The function **using_product_definition_of_item** determines all the **product_definition**s that use the specified **geometric_representation_item** to represent the shape of the **product_definition**s.

EXPRESS specification:

```
*)
-- This function extract a product_definitions using a geometric_representation_item
FUNCTION using_product_definition_of_item
    (item: geometric_representation_item) : SET OF product_definition;
LOCAL
  local_reps: SET OF shape_representation := [];
  local_s_d_r: SET OF shape_definition_representation := [];
  local_p_d_s: SET OF product_definition_shape := [];
  local_s_a: SET OF shape_aspect := [];
  local_p_d: SET OF product_definition := [];
  i : INTEGER;
END_LOCAL;

-- find representations by applying functions defined in Part 43
local_reps := using_representations (item);

-- find shape_definition_representations
REPEAT i:= 1 TO HIINDEX(local_reps);
    local_s_d_r  := BAG_TO_SET (USEDIN (local_reps[i],
                ‘PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.SHAPE_DEFINITION_’ +
                ‘_REPRESENTATION.REPRESENTATION_MODEL’));
END_REPEAT;

-- find product_definition_shape & product_definitions
REPEAT i := 1 TO HIINDEX (local_s_d_r);
   IF ((‘PRODUCT_PROPERTY_DEFINITION_SCHEMA.PRODUCT_DEFINITION_SHAPE’ IN TYPEOF
       (local_s_d_r[i].representation_of)) AND
       (‘PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION’ IN
       (local_s_d_r[i].representation_of\property_definition.definition))) THEN
        local_p_d := local_p_d +
                    local_s_d_r[i].representation_of\property_definition.definition;
   ELSE IF ((‘PRODUCT_PROPERTY_DEFINITION_SCHEMA.SHAPE_ASPECT’ IN TYPEOF
           (local_s_d_r[i].representation_of)) AND
           (‘PRODUCT_DEFINITION_SCHEMA.PRODUCT_DEFINITION’ IN
        (local_s_d_r[i].representation_of.of_shape\property_definition.definition)))
   THEN local_p_d := local_p_d +
                    local_s_d_r[i].representation_of.of_shepe\
                    property_definition.definition;
   END_IF;
   END_IF;
END_REPEAT;
RETURN (local_p_d);
END_FUNCTION;
(*
```

Attribute definitions:
item: the input **geometric_representation_item** used to represent the shape of the **product_definition**s, which shall be extracted.

4.5.6 consistent_of_af_association_and_af_association_rep

The function **consistent_of_af_association_and_af_association_rep** examines the consistency of the **assembly_feature_association** and the **representaing_relationship** which are specified by an **assembly_feature_association_representation**. In other words, it is examined whether the **relating_-shape_aspect** of the **assembly_feature_association** is represented by the **relating_representation** or **the relating_representation_item** of the relating element of the **representing_relationship**, and whether the **related_shape_aspect** of the **assembly_feature_association** is represented by the **related_-representation** or **the related_representation_item** of the relating element of the **representing_-relationship**

EXPRESS specification:

```
*)
-- This function examines the consistency assembly_feature_association and
-- representaing_relationship
FUNCTION consistent_of_af_association_and_af_association_rep
    (item:assembly_feature_association_representation) :LOGICAL;
LOCAL
  local_relating_sa: shape_aspect;
  local_related_sa: shape_aspect;
  local_relating_rep: representation;
  local_related_rep: representation;
  local_relating_rep_item: representation_item;
  local_related_rep_item: representation_item;
END_LOCAL;

-- find relating and related shape aspects
local_relating_sa := item.definition.relating_shape_aspect;
local_related_sa := item.definition.related_shape_aspect;

-- find relating and related representation_items and examine consistency
-- if item uses parametric_assembly_constraint
IF ('PARAMETRIC_ASSEMBLY_CONSTRAINTS_SCHEMA.PARAMETRIC_ASSEMBLY_CONSTRAINT' IN
    TYPEOF (item.representation_relation)) THEN
    IF (SIZEOF (item.representation_relation\explicit_geometric_constraint.
                reference_element) :=: 2) THEN
      local_relating_rep_item := item.representation_relation\explicit_\
                                 geometric_constraint.reference_element[1];
      local_related_rep_item := item.representation_relation\explicit_\
                                 geometric_constraint.reference_element[2];
    ELSE
      local_relating_rep_item := item.representation_relation\explicit_\
                                 geometric_constraint.reference_element[1];
      local_related_rep_item := item.representation_relation\explicit_\
                                 geometric_constraint.constrained_element[1];
    END_IF;
    IF ((local_relating_sa IN using_shape_aspect_of_item
                              (local_relating_rep_item)) AND
        (local_related_sa IN using_shape_aspect_of_item
                              (local_related_rep_item))) THEN RETURN (TRUE);
    ELSE RETURN (FALSE);
    END_IF;
```

```
-- if item uses representation_relationship_with_motion
ELSE IF ('REPRESENTATION_RELATIONSHIP_WITH_MOTION' IN
          TYPEOF (item.representation_relation)) THEN
        local_relating_rep := item.representation_relation
             \representation_relationship.relating_representation;
        local_related_rep_item := item.representation_relation
             \representation_relationship.related_representation ;
        IF ((local_relating_sa IN using_shape_aspect_of_representation
                               (local_relating_rep)) AND
            (local_related_sa IN using_shape_aspect_of_representation
                               (local_related_rep_item))) THEN RETURN (TRUE);
        ELSE RETURN (FALSE);
        END_IF;

-- if item uses kinematic_pair
ELSE IF ('KINEMATIC_STRUCTURE_SCHEMA.KINEMATIC_PAIR' IN
          TYPEOF (item.representation_relation)) THEN
         local_relating_rep_item := item.representation_relation
             \item_defined_transformation.transform_item_1;
         local_related_rep_item := item.representation_relation
             \item_defined_transformation.transform_item_2;
        IF ((local_relating_sa IN using_shape_aspect_of_item
                               (local_relating_rep_item)) AND
            (local_related_sa IN using_shape_aspect_of_item
                               (local_related_rep_item))) THEN RETURN (TRUE);
        ELSE RETURN (FALSE);
        END_IF;
ELSE RETURN (FALSE);
END_IF;
END_IF;
END_IF;
END_FUNCTION;
(*
```

Attribute definitions:

item: the input of an **assembly_feature_association_representation** whose consistency should be examined.

4.5.7 using_shape_aspect_of_representation

The function **using_shape_aspect_of_representation** determines all the **shape_aspect**s that is represented by the specified **representation**.

EXPRESS specification:

```
*)
-- This function extract a set of shape_aspect using a representation
FUNCTION using_shape_aspect_of_representation
    (item: representation) : SET OF shape_aspect;
LOCAL
  local_s_d_r: SET OF shape_definition_representation := [];
  local_s_a: SET OF shape_aspect := [];
  i : INTEGER;
END_LOCAL;
-- find shape_definition_representations
 local_s_d_r  := BAG_TO_SET (USEDIN (item,
                'PRODUCT_PROPERTY_REPRESENTATION_SCHEMA.SHAPE_DEFINITION_' +
                '_REPRESENTATION.REPRESENTATION_MODEL'));
-- find shape_aspects
REPEAT i := 1 TO HIINDEX (local_s_d_r);
```

```
   IF ('PRODUCT_PROPERTY_DEFINITION_SCHEMA.SHAPE_ASPECT' IN TYPEOF
           (local_s_d_r[i].representation_of))
      THEN local_s_a := local_s_a + local_s_d_r[i].representation_of;
   END_IF;
END_REPEAT;
RETURN (local_s_a);
END_FUNCTION;
(*
```

Attribute definitions:
item: the input of a **representation** used to represent the **shape_aspect**s, which shall be extracted.

4.5.8 using_shape_aspect_of_item

The function **using_shape_aspect_of_item** determines all the **shape_aspect**s that is represented by the **representation**s including the **specified representation_item**.

EXPRESS specification:

```
*)
-- This function extract a shape_aspect using a representation_item
FUNCTION using_shape_aspect_of_item
    (item: representation_item) : SET OF shape_aspect;
LOCAL
  local_reps: SET OF shape_representation := [];
  local_s_d_r: SET OF shape_definition_representation := [];
  local_s_a: SET OF shape_aspect := [];
  i : INTEGER;
END_LOCAL;

-- find representations by applying functions defined in Part 43
local_reps := using_representations (item);
-- find shape_aspect
REPEAT i := 1 TO HIINDEX (local_s_d_r);
   IF ('PRODUCT_PROPERTY_DEFINITION_SCHEMA.SHAPE_ASPECT' IN TYPEOF
           (local_s_d_r[i].representation_of))
      THEN local_s_a := local_s_a + local_s_d_r[i].representation_of;
   END_IF;
END_REPEAT;
RETURN (local_s_a);
END_FUNCTION;

END_SCHEMA;  -- assembly_model_schema
(*
```

Annex A
(normative)
Short names of entities

(to be completed)

Annex B
(normative)
Information object registration

(to be completed)

Annex C
(informative)
Computer-interpretable listings

(to be completed)

Annex D
(informative)
EXPRESS-G figures

The EXPRESS-G representation for the schema defined in the subclauses 4 of this part of ISO 10303 are provided in the following figures.

Note: All the entities related with the assembly_model_schema is also included in the figure for the easy understanding of the model.

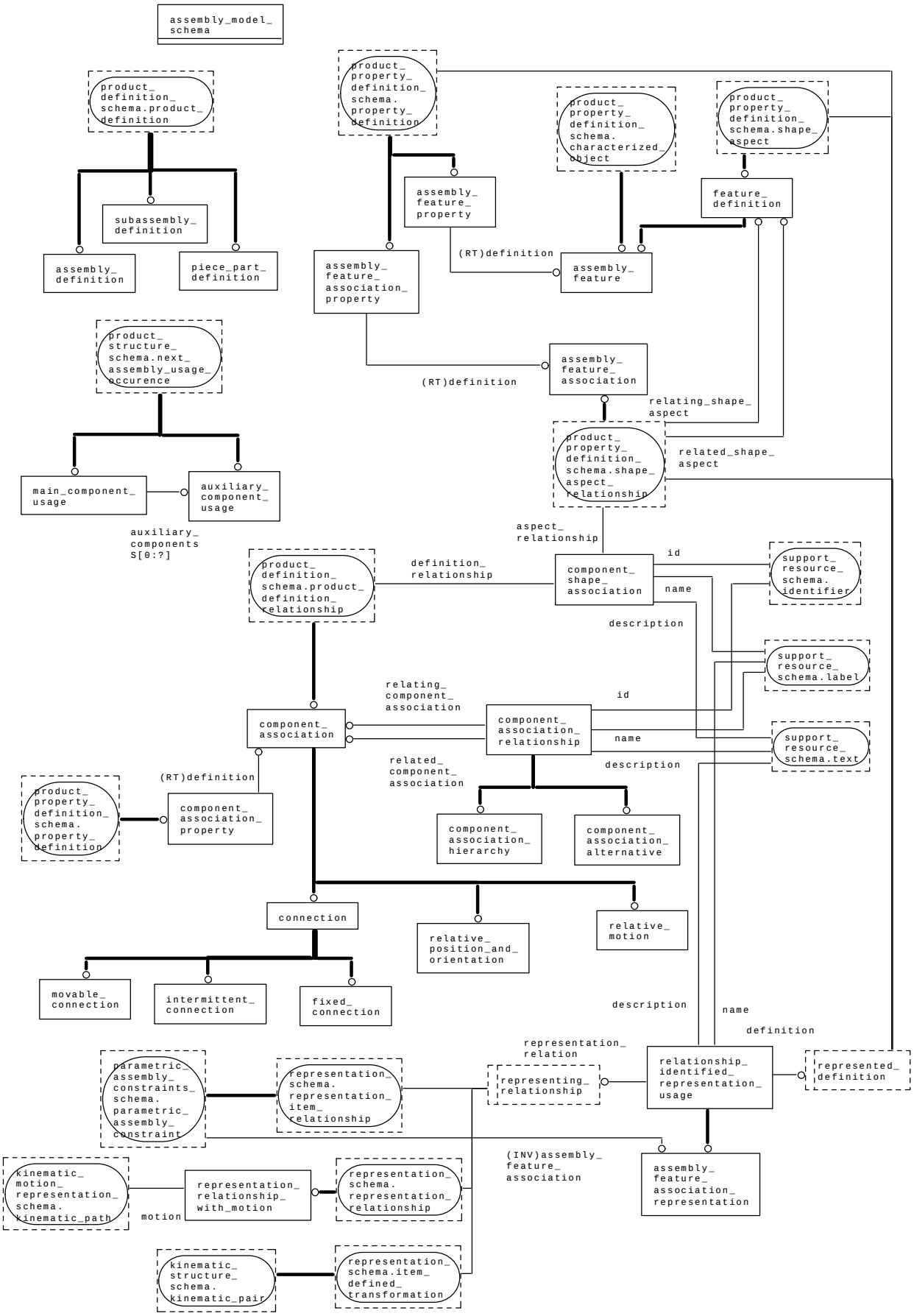


Fig. 1 EXPRESS-G diagram of assembly_model_schema

Annex E
(informative)
Explanation of Assembly Model of Products

1. INTRODUCTION

The objective of this attachment is to present a basic idea and examples of the product model called an assembly model, which is applicable to describe the assemblies based on the STEP framework. In particular, emphasis is given to establish a data structure to represent the peer to peer associations among the components of the assemblies.

2. CONTENTS OF ASSEMBLY MODEL

2.1 Target Product

The target products to be represented by the assembly model are summarized as follows.

(1) products composed of sets of components.

The products considered here are the assembled products composed of sets of the components. The whole products are called "assemblies", and the components of the lowest levels in the assemblies are called "parts". The components of the intermediate levels are called "sub-assemblies", which are composed of one or more parts and/or sub-assemblies. An assembly consists of one or more sub-assemblies and parts.

(2) Product structure configuration of assembly

Product structure configuration is now dealt with in ISO 10303 - 44 to describe the parts lists and the BOM (bill-of-material). Various structure configurations are given to one assembly depending on various contexts. For instance, one configuration of an assembly is considered in the design phase, and the structure configuration may be changed in the assembly process planning phase. The objective of the assembly model is to establish a model describing both the product structure configuration and the connecting associations among the components needed in the various design, analysis and manufacturing process planning phases.

(3) Standard parts

The assemblies include many standard parts, such as fixing bolts, keys, and electric motors. The standard parts are basically divided into two; they are, the standard parts included in the parts catalogues discussed in ISO TC 184/SC4/WG2, and the standard parts defined by the users.

2.2 Interest Application Fields

The interest application fields of the assembly model are as follows.

(1) Kinematic analysis of assemblies

Kinematic analysis and simulation are very important application fields of the assembly model. ISO 10303-105 Kinematics supports the kinematic analysis, however, Part 105 is not sufficient to integrate the 3D-CAD systems and the kinematic analysis systems.

(2) Animation of assemblies

The animation of assemblies is very important for future extension of the digital-mockup technologies.

(3) Assembly/disassembly process planning

The assembly model will support the integration of the product design and the manufacturing preparation. The assembly process planning and the disassembly process planning are important application fields of the assembly model from the viewpoint of the integration of CAD and CAM systems.

(4) Tolerance analysis and synthesis

The tolerance analysis and synthesis of the complicated assembly are very important application fields of the assembly model.

2.3 Contents of Assembly Model

The contents of the assembly model are analyzed from the viewpoints of the design, the analysis and the manufacturing preparation of the products. The contents are classified into four classes.
They are,

(1) Information of individual parts.
(2) Information of standard parts.
(3) Structure configuration of assembly
a) Hierarchical associations (parent-child associations) among assemblies, subassemblies and parts.
b) Positions and orientations of components in a higher level component.
c) Tolerance of the positions and orientations
(4) Component Association
a) Peer to peer associations among components.
b) Relative positions and orientations of components against other components.
c) Relative motions of components against other components.
d) Tolerance of the relative motions, positions and orientations.
e) Assembly features needed to define technological information of component associations.

Table 1 Contents of STEP model

Information	STEP Model
(1), (3)-a), (3)-b) (1) (3)-b) (3)-a), (3)-b) (1) (1), (3)-a), (3)-b) (1) (1), (3)-a), (3)-b) (1) (4)-a), b), c) (2)	ISO 10303: Part 41 : Fundamentals of Product Description and Support. ISO 10303: Part 42 : Geometric and Topological Representation. ISO 10303: Part 43 : Representation Structures. ISO 10303: Part 44 : Product Structure Configuration. ISO 10303: Part 45 : Materials. ISO 10303: Part 46 : Visual Presentation. ISO 10303: Part 47 : Shape Tolerances. ISO 10303: Part 49 : Process Structure and properties. ISO 10303: Part 101: Draughting. ISO 10303: Part 104: Finite Element Analysis. ISO 10303: Part 105: Kinematics. ISO 13584: Standard Parts

Most of the information mentioned above can be described by the integrated generic resources and application resources defined in the Parts of the STEP (ISO 10303). Table 1 summarizes the contents of the STEP model.

The information about the individual parts are represented by ISO 10303-41, 42, 43, 45, 46, 47 and 49., and the information about the standard parts may be defined by ISO 13584.

As regards the assembly information (3) and (4), ISO 10303-44 Product Structure Configuration model provides a mechanism to represent (3)-a) and (3)-b). ISO 10303-105 Kinematics gives the mechanism to represent the relative motion between pairs of links that are a set of components fixed with each other and represent kinematic aspects of (4)-a), b) and c).

Therefore, the assembly model deals with the following items of the products.

(4) Component Association
- a) Peer to peer associations among components.
- b) Relative positions and orientations of components against other components.
- c) Relative motions of components against other components.
- e) Assembly features needed to define technological information of component associations.

The following items related to the tolerances are considered in the other Part of this Standards.

(3) Structure configuration of assembly

c) Tolerance of the positions and orientations

(4) Component Association

d) Tolerance of the relative motions, positions and orientations.

3. BASIC IDEA OF ASSEMBLY MODEL

3.1 Assembly Features

Figure 1 (a) shows a simple example of the assembly, which consists of two plates and a fixing bolt. In this case, the connecting associations among the components are presented in the diagram of Fig. 1 (b). In the figure (b), the rectangular blocks show the shape elements on which a pair of components are connected and/or associated. The shape elements, such as planes, holes, screws and nuts, are called “assembly feature” in the assembly model.

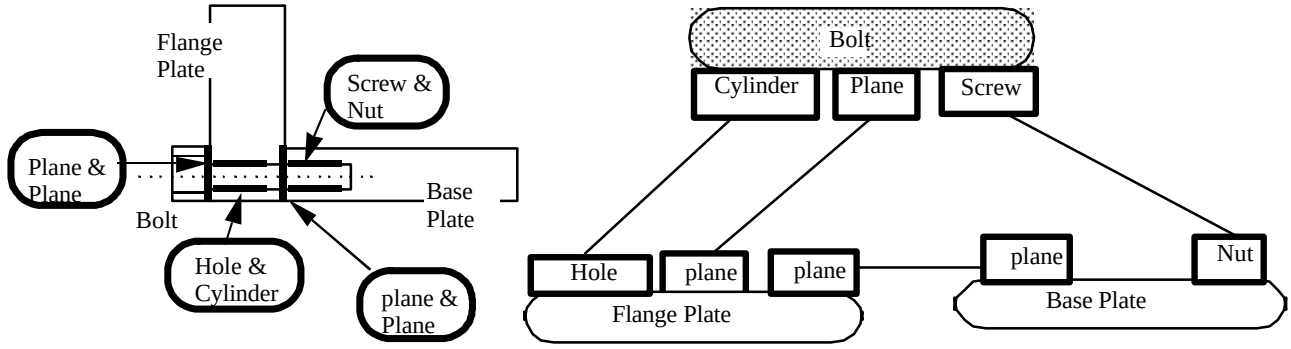
The assembly features are the elemental entities for representing the peer to peer associations between a pair of components. For the example, two cases may be considered to represent the assembly features between the flange plate and the bolt in Fig. 1(b) .

CASE1: The each individual component has one assembly feature.
Assembly feature 1 of flange plate = Hole + Plane
Assembly feature 2 of bolt = Cylinder + Plane

CASE 2: The individual faces are defined as the individual features.
Assembly feature 11 of flange plate = Hole
Assembly feature 12 of flange plate = Plane
Assembly feature 21 of bolt = Cylinder
Assembly feature 22 of bolt = Plane

The definition of the assembly feature depends on the various viewpoints and the application fields, therefore, both the assembly feature definition of CASE 1 and CASE 2 should be taken into consideration in the assembly model.

Figure 2 shows another example of the assembly of the automotive engines. In this case, the connecting associations include both the fixed connecting association and the movable connecting associations, as shown in Fig. 3. The types of the joints are described for the individual association between the pairs of the assembly features in Fig. 3. A spring part, an elastic component, is also included in the assembly.



(a) Drawing (b) Connecting association among components

Fig. 1 Assembly model example 1: Two plates are fixed by a bolt

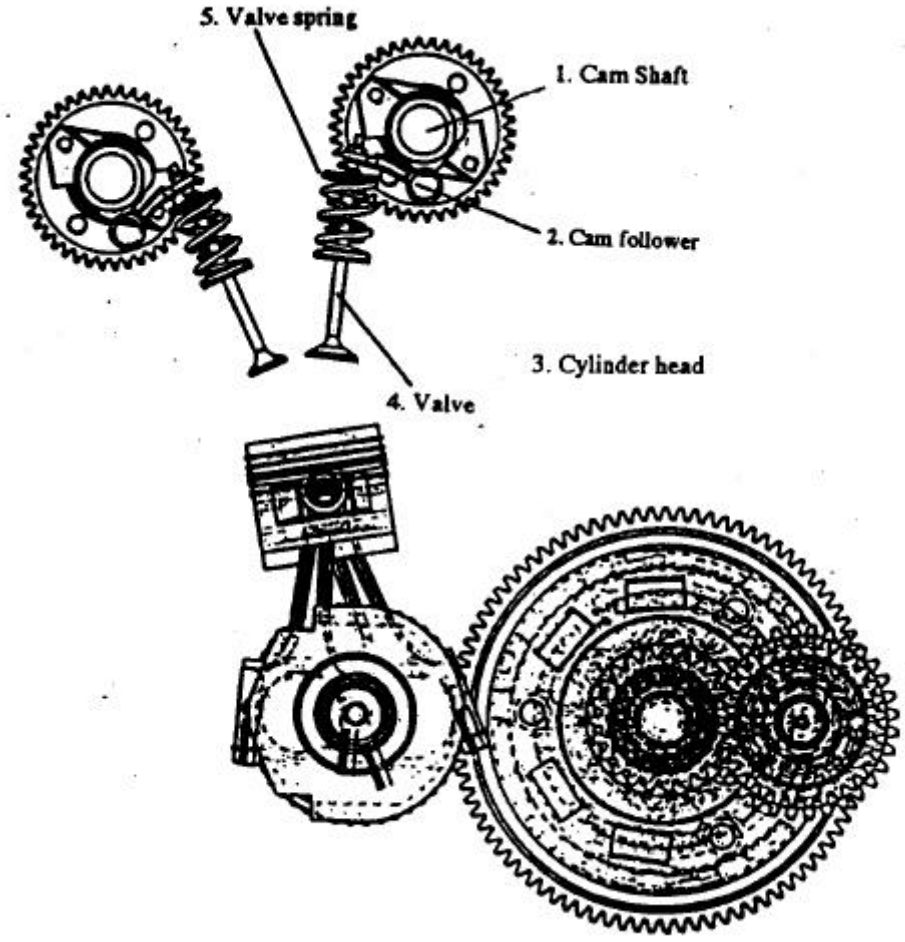


Fig. 2 Assembly model example 2: Automotive engine

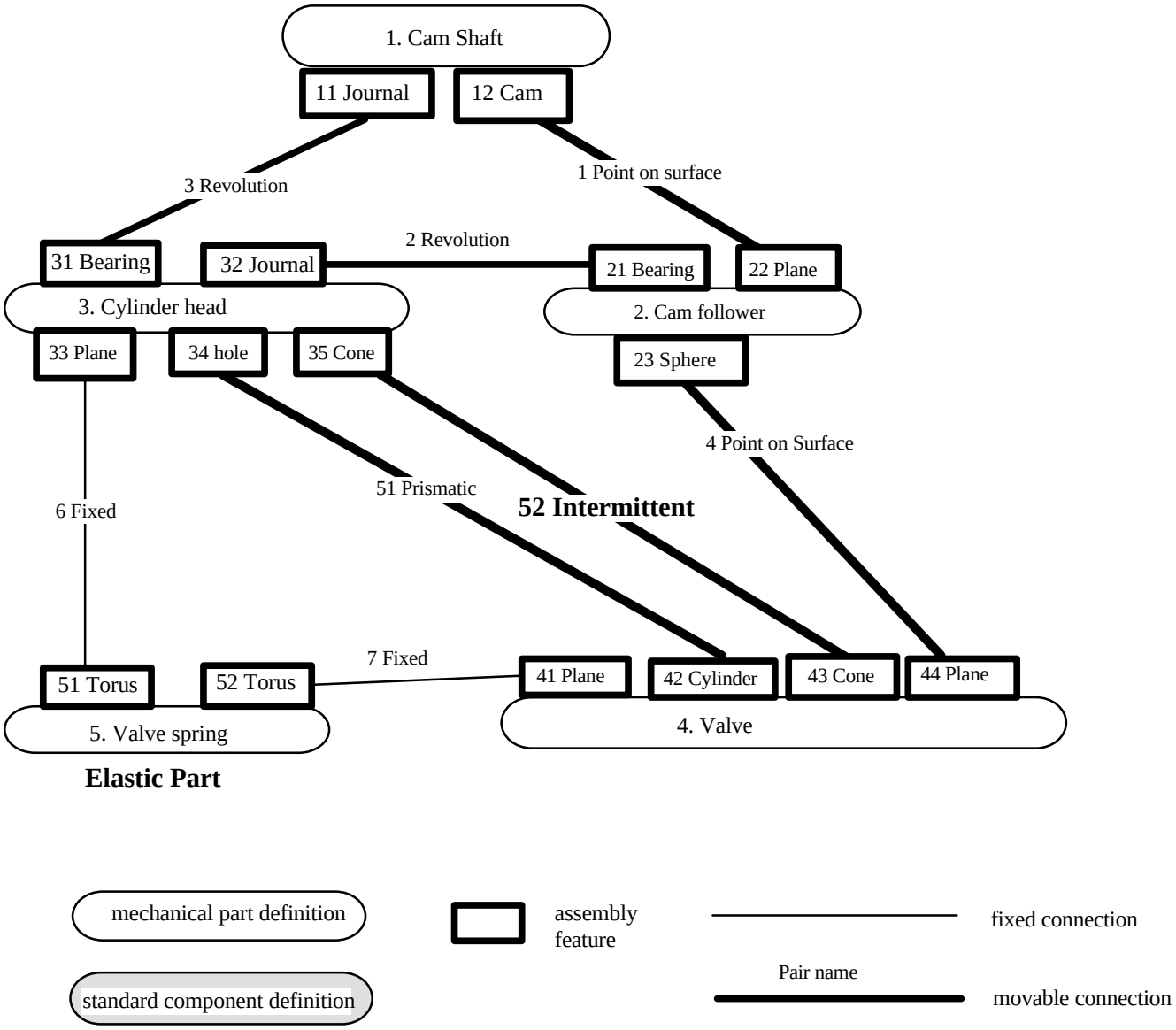


Fig. 3 Connecting associations of assembly model example 2

3.2 Basic Structure of Assembly Model

Basic structure of the assembly model considered here is shown in Fig, 4. In the figure, the boxes and the circles describe the components (assemblies, subassemblies, and parts) and the hierarchical associations (parent-child associations) among the component definitions, respectively.

The hierarchical associations are classified into two types.
(1) main_component_usage
This entity represents the parent-child associations between a pair of components, which are main and/or important components.

(2) auxiliary_component_usage
This entity represents the parent-child association between a pair of components, which are additional, and/or auxiliary components such as fixing bolts, rivets and so on.

In the case of the assembly shown in Fig. 1 (b), the hierarchical associations are as follows.

(a) Assembly ← **main_component_usage** → Flange plate
(b) Assembly ← **main_component_usage** → Base plate
(c) Assembly ← **auxiliary_component_usage** → Bolt

The solid circles in Fig. 4 show the components associations (peer to peer associations) among the components (assemblies, subassemblies, and parts). For examples, all the arcs in Fig. 1 (b) and Fig. 3 represent the peer to peer associations. The details of the components associations are described by the assembly feature associations and the assembly features.

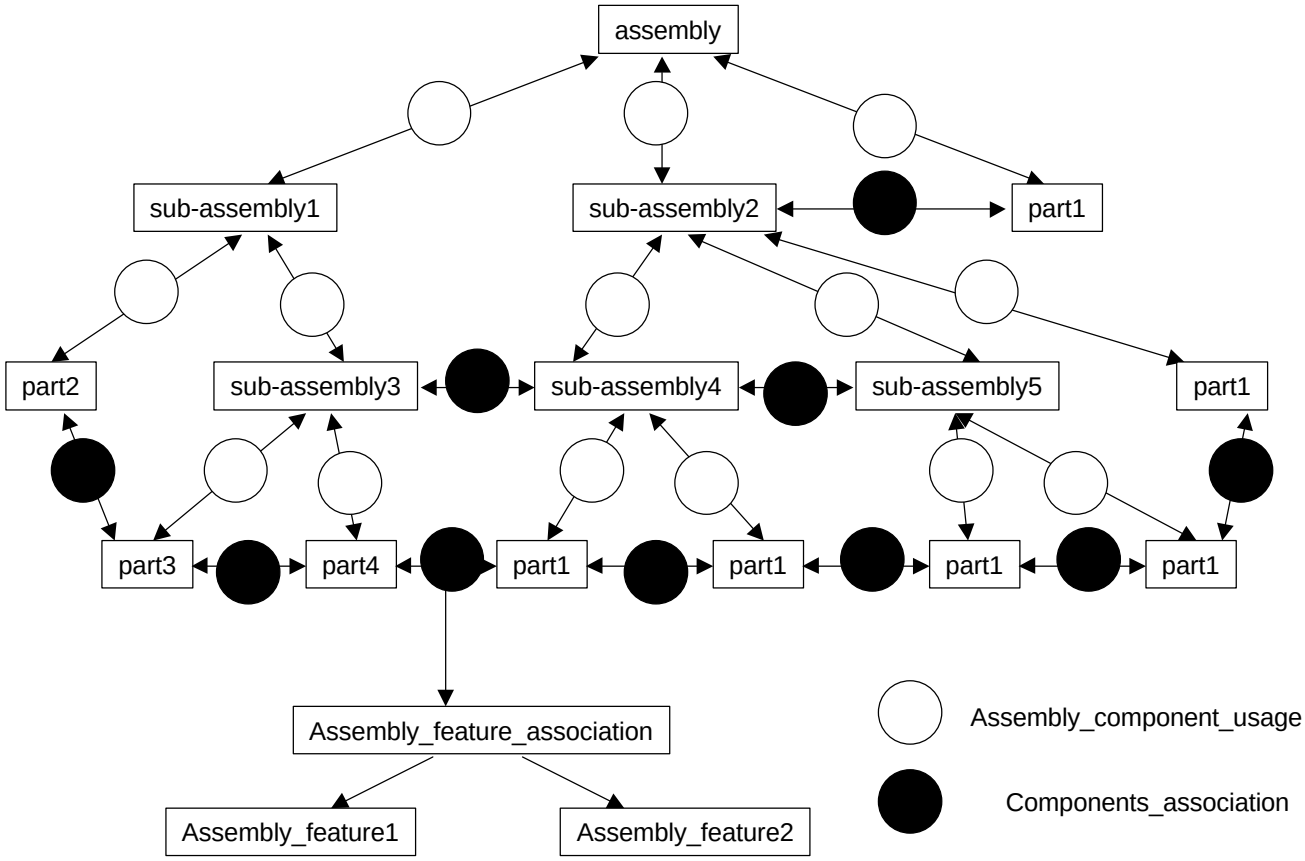


Fig. 4 Basic structure of assembly model

Index

(to be completed)